Serial No.: 09/888,273
Amendment and RCE dated: January 8, 2007
Advisory Action dated: August 28, 2006

REMARKS/ARGUMENTS

Claims 1, 2, 4-18, 20-26 and 28-30 are pending. Claims 3, 19 and 27 were

previously cancelled. Claims 1, 2, 7, 9-11, 15, 17, 18, 23, 25, and 26 are rejected under

35 U.S.C. §102(e) as being anticipated by Sager, US Patent 6,542,921 ("Sager").

Applicants gratefully acknowledge the Office Action's indication that claims 4-6, 8, 12-

14, 16, 20-22, 24 and 28-30 contain allowable subject matter. *See* Office Action dated

4/5/2006, paragraph 7. Claims 1, 5, 7, 9, 12, 15, 17, 23, 25 and 29 are amended. Claims

11 is cancelled without prejudice or disclaimer.

Applicants respectfully submit nowhere in Sager is the disclosure or teaching of

"[a] method of assigning thread priority comprising ... determining if there is an

indication of approaching instruction side starvation for said first thread wherein

instruction fetching for said first thread would be blocked due to processing one or more

instructions from another thread; and incrementing a value stored in said first starting

counter in response to an indication of approaching instruction side starvation for said

first thread" (*e.g.*, as described in the embodiment of claim 1).

First, the Office Action asserts Sager has taught determining if there is an

indication of approaching instruction side starvation for said first thread in the Abstract

and the description of Figure 11, element 1117. *See* Office Action dated 9/19/2005,

paragraph 4d. Applicants disagree. The Abstract states:

> The present invention provides a method and apparatus for controlling a
> processing priority assigned alternately to a first thread and a second
> thread in a multithreaded processor to prevent deadlock and livelock
> problems between the first thread and the second thread. In one
> embodiment, the processing priority is initially assigned to the first thread
> for a first duration. It is then determined whether the first duration has

expired in a given processing cycle. If the first duration has expired, the processing priority is assigned to the second thread for a second duration.

The Abstract merely describes a general intention to address deadlock and livelock problems between a first and second thread. It further describes that when a first duration of processing priority is expired, a second duration to a second thread is assigned. There is no mention of at least "...determining if there is an indication of approaching instruction side starvation for said first thread wherein instruction fetching for said first thread would be blocked due to processing one or more instructions from another thread ...", as described in the embodiment of claim 1.

Next, the description of Element 1117 states:

At decision block 1117, the process proceeds to block 1121 if it is determined that the processing priority has been switched from thread 0 to thread 1 in the current processing cycle and loops back to block 1105 otherwise. In one embodiment, whether the processing priority has been switched back from thread 0 to thread 1 in the current processing cycle can be determined by detecting a signal indicating that the content of the TPC has reached the predetermined threshold value in the current processing cycle and that the TPB has been inverted from 0 to 1 in the current processing cycle. The determination of whether the processing priority has been switched from thread 0 to thread 1 in the current processing cycle will be described in more detail below.

The above cited section describes the switching of a thread 0 to a thread 1 conditioned upon the detecting a signal indicating a predetermined threshold value and another the inversion of another value. However, again there is no mention of at least the specifically claimed limitation "...determining if there is an indication of approaching instruction side starvation for said first thread wherein instruction fetching for said first thread would be blocked due to processing one or more instructions from another thread...", as described

Serial No.: 09/888,273
Amendment and RCE dated: January 8, 2007
Advisory Action dated: August 28, 2006

in the embodiment of claim 1. Applicants submit that these cited sections do not address Iside starvation or incrementing values at all, and that the Office Action's assumption regarding the operation of Sager are unsupported by the text of Sager reference.

In addition, the Office Action cites the extensive section column 9, line 3 – column 10, line 26 of Sager as allegedly describing the relevant limitations. *See* Office Action dated 4/5/2006, page 3. Applicants respectfully disagree. The first paragraph of this section describes a flexible and alternating priority scheme in which each thread is alternately given the priority for a period of time to progress, thereby enabling other threads to progress as well. The second paragraph further describes the timing mechanism of this priority scheme, stating a thread is not to be given priority under any circumstances, even if the thread is "stuck". The third paragraph describes a thread precedence bit (TPB) used to indicated which of a number of threads has priority at a given moment. The fourth paragraph describes the parameters of thread "progress"; specifically, a thread makes progress if it has no instructions to be retired, or if it has retired at least one instruction in a current processing period. The final fifth paragraph describes in the Sager process in step-wise form. Applicants submit the cited reference does not describe at least "...determining if there is an indication of approaching instruction side starvation for said first thread wherein instruction fetching for said first thread would be blocked due to processing one or more instructions from another thread ..." anywhere.

Applicants further submit the Office Action's assertion Sager's teaching that when a thread of instructions has not made any progress during its assigned priority time

96006_1.doc   -13-

Serial No.: 09/888,273
Amendment and RCE dated: January 8, 2007
Advisory Action dated: August 28, 2006

period, the next time the thread executes it will be given a longer priority period to

execute is the same as the relevant limitation is incorrect. *See* Office Action dated

4/5/2006, page 3. An execution process that simply calls for a longer priority period

simply because one thread has not made any progress is not the same as a determination

of instruction side starvation. In order to support a proper rejection, the Sager reference

must teach "...determining if there is an indication of approaching instruction side

starvation for said first thread wherein instruction fetching for said first thread would be

blocked due to processing one or more instructions from another thread...". Since the

cited sections of the Sager reference do not describe starvation at all, the current rejection

is inadequate.

Furthermore, Applicants submit the cited reference does not teach

"...incrementing a value stored in said first starting counter in response to an indication

of approaching instruction side starvation for said first thread ...", as described in the

embodiment of claim 1. The Office Action cites Figure 11, element 1121 and column 9,

lines 12-29 in its rejection. *See* Office Action dated 9/19/2005, paragraph 4e.

The description of element 1121 states:

> At block 1121, the TC0 content is incremented by a predetermined number, for
> example 1. The content of the TC0, as explained above, will be used to load into
> the TPC to indicate how long the priority duration for thread 0 will be the next
> time thread 0 is given the processing priority.

Applicants submit that the cited section describes the incrementing of TC0 content.

However, column 12 lines 1-3 state:

> T0 counter (TC0) is used to hold a value that corresponds to a duration for which
> the thread 0 is given the processing priority.

96006_1.doc   -14-

Serial No.: 09/888,273
Amendment and RCE dated: January 8, 2007
Advisory Action dated: August 28, 2006

It is clear that the incrementing of TC0 counter found in the cited section above has nothing to do with approaching iside starvation, but rather to determining *the duration of time* for which a thread is to be give processing priority. Next, column 9, lines 12-29 state:

> As each thread is being executed, its progress is monitored to determine whether it is being stuck. *If a particular thread, for example thread 0, has not made any progress in the period of time during which it has priority then it will be given priority for a longer duration of time the next time it has priority.* This duration of time during which thread 0 is given priority will continue to increase until thread 0 makes progress. Once it is determined that thread 0 has made progress, its priority duration can be reset to some shorter period, for example the initial duration. Likewise, the duration of time during which thread 1 is given priority will continue to increase until thread 1 makes some progress at which time its priority duration can be reset to some shorter period, for example the initial duration. *In short, the length of time during which each thread has priority will continue to increase until that particular thread makes some progress. (emphasis added)*

Applicants submit this section reflects the operation of the TC0 counter described above, in that it addresses the appropriate amount of time a thread is to be given priority, to be determined based upon the amount of progress the given thread has made. Again, this section does not address "...incrementing a value stored in said first starting counter in response to an indication of approaching instruction side starvation for said first thread ...", as described in the embodiment of claim 1.

The Office Action also cites to column 9, line 3 – column 10, line 26 and column 13, lines 42-51 of Sager as allegedly describing the relevant limitations. *See* Office Action 4/5/2006, page 3. However, for similar reasons to those discussed above, the extensive section of column 9, line 3 – column 10, line 26 fails to describe iside starvation at all. Applicants submit column 13, lines 42-51, further describes the

96006_1.doc   -15-

Serial No.: 09/888,273
Amendment and RCE dated: January 8, 2007
Advisory Action dated: August 28, 2006

operation of the TC0 counter (described above), specifically directed toward "how long the priority duration for thread 0 will be the next time thread 0 is given the processing priority", and has nothing to do with instruction side starvation. *See* column 13, lines 48-51. Applicants respectfully submit since the cited reference does not disclose multiple limitations as described in the embodiment of claim 1, it is inadequate to support a proper 35 U.S.C. §102(e) rejection.

With regard to the rejection of independent claim 7, Applicants' respectfully submit that nowhere is the disclosure or teaching of "[a] method of assigning thread priority comprising ... assigning priority to a second thread in response to one of a plurality of conditions being true, the conditions consisting of... if there is an indication of approaching instruction side starvation for said first thread wherein instruction fetching for said first thread would be blocked due to processing one or more instructions from another thread" (*e.g.*, as described in the embodiment of claim 7).

The Office Action asserts that Sager has taught the claimed limitations at Figure 9 (elements 913 and 917). *See* Office Action dated 9/19/2005, paragraph 9iii. The Office Action further states that when the priority of the second thread is assigned in response to only the current priority period expiring, this means that the condition is true that there is not an indication of approaching instruction side starvation for said thread. Applicants disagree.

First, Applicants first reiterate all arguments made above regarding Iside starvation. Moreover, Applicants' respectfully submit that Office Action's assumptions regarding Sager's disclosure of assigning priority based on a plurality of conditions

96006_1.doc   -16-

Serial No.: 09/888,273
Amendment and RCE dated: January 8, 2007
Advisory Action dated: August 28, 2006

including indication of approaching Iside starvation are both unsupported and not taught

or disclosed in the Sager reference. The description of elements 913 and 917 of Sager

merely state:

> At decision loop 913, the method 900 proceeds to block 917 if the current
> priority duration has expired. At block 917, the processing priority is alternated,
> *i.e.*, assigned to the other thread.

The cited section merely describes the operation of priority designation; priority is

assigned to the other thread when the priority duration of the current thread is expired.

Similar to above, this section Sager reference is merely describing the shifting of a

priority designation between two threads based on a predetermined timing criteria.

It is clear Sager does not disclose assigning priority based on a plurality of

conditions including no indication of approaching Iside starvation (as described in the

embodiment of claim 7) in its description of elements 913 and 917. The Office Action's

assumptions are insufficient and must be found in the reference to form the basis of a

proper 35 U.S.C. §102(e) rejection.

The Office Action asserts when the priority to a second thread is assigned in

response to only current priority period expiring, element 913 (*i.e.*, when progress has

been made during thread execution priority period) the condition is true there is no

indication of approaching instruction side starvation for a thread. *See* Office Action

dated 4/5/2006, paragraph 6. Applicants submit this is improper. To assume simply

because a priority designation is shifted from one thread to another, that there must have

not been an indication of approaching instruction side starvation when there is no citation

to any description of instruction side starvation in the Sager reference anywhere is

improper, self-serving and inadequate to support a proper rejection. In order to support a

96006_1.doc   -17-

Serial No.: 09/888,273
Amendment and RCE dated: January 8, 2007
Advisory Action dated: August 28, 2006

proper rejection, the cited reference must show where instruction side starvation is described, accounted for and executed in the Sager reference.

In order to be a proper 35 U.S.C. §102(e) rejection, the claimed limitations must be taught or disclosed in the cited reference. For at least the reasons discussed above, they are not. Independent claims 7, 9, 15, 17, 23 and 25 contain similar limitations, and therefore are allowable as well. Dependent claims 2-4-6, 8, 10, 12-14, 16, 18, 20-22, 24, 26 and 28-30 depend from allowable base claims and therefore should be allowed as well.
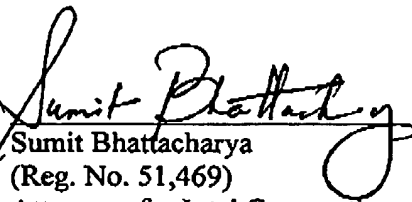
For at least all the above reasons, the Applicants respectfully submit that this application is in condition for allowance. A Notice of Allowance is earnestly solicited.

The Examiner is invited to contact the undersigned at (408) 975-7500 to discuss any matter concerning this application. The Office is hereby authorized to charge any additional fees or credit any overpayments under 37 C.F.R. § 1.16 or § 1.17 to Deposit Account No. 11-0600.

Respectfully submitted,
KENYON & KENYON LLP

Dated: January 8, 2007   By: Sumit Bhattacharya
                                 Sumit Bhattacharya
                                 (Reg. No. 51,469)
                                 Attorneys for Intel Corporation

KENYON & KENYON LLP
333 W. San Carlos St., Suite 600
San Jose, CA 95110

Telephone: (408) 975-7500
Facsimile: (408) 975-7501

96006_1.doc   -18-